

# DBMS-KU at SemEval-2019 Task 9: Exploring Machine Learning Approaches in Classifying Text as Suggestion or Non-Suggestion

Tirana Noor Fatyanosa<sup>1</sup>, Al Hafiz Akbar Maulana Siagian<sup>1,3</sup>, Masayoshi Aritsugi<sup>2</sup>

<sup>1</sup>Computer Science and Electrical Engineering

Graduate School of Science and Technology, Kumamoto University, Japan

<sup>2</sup>Big Data Science and Technology

Faculty of Advanced Science and Technology, Kumamoto University, Japan

<sup>3</sup>Indonesian Institute of Sciences, Indonesia

{fatyanosa,alha002}@dbms.cs.kumamoto-u.ac.jp

aritsugi@cs.kumamoto-u.ac.jp

## Abstract

This paper describes the participation of DBMS-KU team in the SemEval 2019 Task 9, that is, suggestion mining from online reviews and forums. To deal with this task, we explore several machine learning approaches, i.e., Random Forest (RF), Logistic Regression (LR), Multinomial Naive Bayes (MNB), Linear Support Vector Classification (LSVC), Sublinear Support Vector Classification (SSVC), Convolutional Neural Network (CNN), and Variable Length Chromosome Genetic Algorithm-Naive Bayes (VLCGA-NB). Our system obtains reasonable results of F1-Score 0.47 and 0.37 on the evaluation data in Subtask A and Subtask B, respectively. In particular, our obtained results outperform the baseline in Subtask A. Interestingly, the results seem to show that our system could perform well in classifying Non-suggestion class.

## 1 Introduction

Nowadays, a huge number of texts are posted in online reviews or discussion forums. Such media can be a valuable source for obtaining a suggestion about products or services (Negi and Buitelaar, 2015; Negi et al., 2016). The obtained suggestion is not only useful for readers but also important information for stakeholders (Negi et al., 2016). Indeed, such advice can be used to improving the quality of products or giving helpful recommendations (Brun and Hagege, 2013). However, identifying a suggestion from a lot of reviews or comments needs extra effort and time. Moreover, such online texts are mostly in unstructured form (Negi et al., 2018; Negi and Buitelaar, 2017). Thus, automatically mining the suggestion from given texts is challenging and significant (Negi et al., 2016).

Suggestion mining is relatively a new research interest in text classification tasks (Negi and Buitelaar, 2015). Several studies have initiated to mining suggestions from online texts (Negi et al.,

2018; Negi and Buitelaar, 2017; Negi et al., 2016; Negi, 2016; Negi and Buitelaar, 2015; Brun and Hagege, 2013; Ramanand et al., 2010; Dong et al., 2013; Wicaksono and Myaeng, 2012, 2013). Particularly, (Negi and Buitelaar, 2015; Brun and Hagege, 2013; Ramanand et al., 2010) have tried to identify suggestions from customer reviews. Meanwhile, (Negi, 2016; Dong et al., 2013; Wicaksono and Myaeng, 2012, 2013) have mined such advice by using Twitter or discussion forums dataset. Then, (Negi et al., 2018; Negi and Buitelaar, 2017) have utilized WikiHow and open domain corpora for their work. However, they concluded that it is not easy to identify suggestion texts automatically. In other words, it still has room to improving the classification result in the suggestion mining task. The task of suggestion mining from online reviews and forums, namely, Task 9 (Negi et al., 2019), is opened in the International Workshop on Semantic Evaluation 2019 (SemEval-2019).

This paper delineates the participation of DBMS-KU team in both Subtask A and Subtask B of Task 9 of SemEval-2019 (Negi et al., 2019). To address these two Subtasks, we utilize several approaches, namely, Random Forest (RF), Logistic Regression (LR), Multinomial Naive Bayes (MNB), Linear Support Vector Classification (LSVC), Sublinear Support Vector Classification (SSVC), Convolutional Neural Network (CNN), and Variable Length Chromosome Genetic Algorithm Naive Bayes (VLCGA-NB). The obtained results of our experiments are encouraging and show a promising improvement in identifying Suggestion and Non-suggestion.

The rest of this paper is organized as follows. Section 2 explains the problem definition, problem formulation, and dataset. Section 3 presents the tools and libraries used in this work. Section 4 describes our employed methods. Section 5 repre-

sents our experiments that consist of data preprocessing, parameter, and evaluation measurement. Section 6 discusses our obtained results. Finally, we conclude this work in Section 7.

## 2 Problem Definition

Suggestion mining is a binary classification problem. Particularly, suggestion mining is a task that labels sentences as Suggestion or Non-suggestion. However, suggestion sentences can have very broad meaning. Thus, the domain and scope of the suggestion text classification should be described. The Task 9 of SemEval-2019 consists of Subtask A and Subtask B that are classifying a suggestion in intra-domain and cross-domain, respectively (Negi et al., 2019).

### 2.1 Problem Formulation

Suggestion text classification consists of assigning *suggestion, nonsuggestion* to  $(s_i, l_j) \in S \times L$ , where  $S$  is sentences and  $L = [l_1, \dots, l_n]$  is a set of  $n$  predefined labels. Each sentence is classified as Suggestion or Non-suggestion class.

### 2.2 Datasets

Dataset used in Task 9 of SemEval-2019 is divided into training, trial, and evaluation parts (Negi et al., 2019). The dataset consists of three columns: id, sentence, and label (see Table 1). The provided dataset is imbalanced in which, overall, Non-suggestion class is larger than Suggestion one.

## 3 Tools and Libraries

The common classification methods, such as Support Vector Machine, Random Forest Classifier, Linear Regression, and Naive Bayes application are facilitated by the most outstanding library for machine learning, namely, SciKit-Learn (Pedregosa et al., 2011). Correspond to its name, NLTK (Bird et al., 2009) is used as the toolkit for Natural Language Processing (NLP) operations such as tokenization, stemming, metrics, corpus, and classification. Pandas (McKinney, 2010) is chosen as the tools for collection and format the data because of its ease of use. The Keras library (Chollet et al., 2015) that runs on top of TensorFlow (Abadi et al., 2015) is also utilized for building high-level neural networks, i.e., for building the Convolutional Neural Network in this work.

Furthermore, we utilize Seaborn<sup>1</sup> and Matplotlib library (Hunter, 2007) as confusion matrix visualization.

## 4 Classification Methods

This section details the classification methods used in our experiments on Suggestion classification.

### 4.1 Baseline

Baseline method provided by organizer utilizes the suggestion keyword, pattern string, and Part-Of-Speech (POS) Tagger matching. The list of suggestion keywords utilized in the baseline is “suggest”, “recommend”, “hopefully”, “go for”, “request”, “it would be nice”, “adding”, “should come with”, “should be able”, “could come with”, “i need”, “we need”, “needs”, “would like to”, “would love to”, “allow”, and “add”. The baseline method also utilizes the wishes identification pattern string from (Goldberg et al., 2009). The POS tag of each word in the sentences is also done to collect Modal and Verb POS tag only. The classification is done by checking all words in the sentence. If the sentence contains one of the three matches, then the sentence is classified as a Suggestion class.

### 4.2 Common Classification Methods

Common classification methods, such as Support Vector Machine (SVM), Random Forest Classifier (RF), Linear Regression (LR), and Naive Bayes (NB) are employed in this research. Two types of SVM are utilized, that is, Linear Support Vector Machine Classifier (LSVC) and Sublinear Support Vector Machine Classifier (SSVC). The implementation of the common classification methods is available at (Fatyanosa, 2019c).

### 4.3 Variable Length Chromosome Genetic Algorithm-Naive Bayes

Variable Length Chromosome Genetic Algorithm - Naive Bayes (VLCGA-NB) is utilized for features selection. We follow the model and parameter from (Fatyanosa et al., 2018). The first step of VLCGA-NB is selecting initial features from keywords that appear in the Suggestion sentences but do not appear in the Non-suggestion ones in the training data. Within the randomly determined maximum chromosome size, these keywords are

<sup>1</sup><https://seaborn.pydata.org/generated/seaborn.heatmap.html>

Table 1: Dataset example

ID	Sentence	Label
9636	Make ProgressRing control available for Windows Phone just like Win8.	1
9706	Either one has to use .NET to access a library or Microsoft advises to do a .NET app with native code in a WinRT component.	0
9709	Don't limit us artificially just because you don't like native developers.	0
9735	These page templates should be updated to use the medium-sized semibold font by default for the title text.	1
...	...	...

then randomly selected as genes in Genetic Algorithm (GA). Therefore, each chromosome within population will have different length with different genes. All populations resulting from the initialization then evolve through generation by passing the crossover, mutation, and selection operator. A number of children produced by crossover and mutation operator are based on Crossover Rate (CR) and Mutation Rate (MR). Two types of crossover are utilized in this research, viz., Union Crossover and Intersection Crossover. The mutation is done by changing the genes with another feature which is not contained in the chromosome. The gene which will be mutated within chromosome is selected by comparing the generated random value with the MR. If the random value is higher than MR, then the gene will be mutated. The purpose of the crossover operator is to help the algorithm to explore the search space, while the purpose of mutation operator is to exploit certain area in the search space. With these operators, there will be diversity within the population that can help to avoid early convergence. By ranking the fitness value using elitist selection, the next population for the next generation is selected from the prior population and the produced children. Only the chromosome with the highest ranking within the number of population will be selected. All these operators are then iterated until the maximum number of generations. The best chromosome produced in the last generation is then used as the Suggestion keywords in the baseline code provided by the organizer. GA, which is a well-known evolutionary algorithm, is one of the powerful stochastic and heuristic algorithms. The use of GA is legion as it can provide search space exploration through crossover operator and exploitation through mutation operator. Thus, GA is possible to search in a very wide search space and allow it to produce nearly optimal results. This ability becomes the motivation for feature selection

using GA. However, the drawback of the GA is that it is not guaranteed to produce the global optimal, but instead satisfactory results. Moreover, GA requires parameter tuning to find the appropriate parameter based on the dataset and needs a longer runtime. Despite its drawback, we expect that GA can produce a limited number of Suggestion features which has a major contribution to the Suggestion classification in this work. The implementation of the VLPGA-NB is available at (Fatyanosa, 2019b).

#### 4.4 CNN

For our purpose in this work, we follow the Keras model's architecture from (Chollet, 2017) as shown in Figure 1. This architecture tends to obtain high accuracy when applied on the News-group dataset. The text classification using CNN is done in four steps. First, all sentences are converted into word index order. Only 20,000 frequently words with the upper limit length of 1000 words will be considered. Next, 100-dimensional Global Vectors for Word Representation (GloVe) embeddings are utilized as the embedding matrix. Then, this matrix is loaded into Embedding layer of Keras. Finally, the Softmax function is used in the final layer of CNN. The implementation of the CNN is available at (Fatyanosa, 2019a). Although there are several pre-trained word vectors, GloVe and word2vec are considered as the most popular vectors (Lee et al., 2016). Based on (Pennington et al., 2014), their GloVe vector has outperformed other word representations in terms of word comparison, correlation, and named entity recognition. We thus use the GloVe vector as our pre-trained word embedding in this work.

## 5 Experiments

We conducted the experiments with the seven classification methods in this section. We employed the datasets from both Subtasks for evaluating the

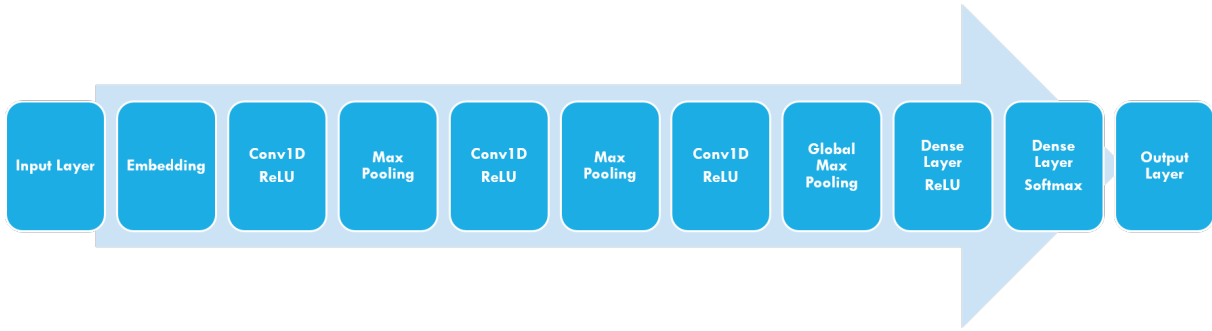


Figure 1: CNN Architecture from (Chollet, 2017)

performance of those classification methods. We compared the performance of seven classification methods against the baseline provided by the organizer.

### 5.1 Data Preprocessing

We performed a sequence of preprocessing steps to address noise in the data. For RF, MNB, LSVM, and LR, all features were converted into numerical feature vectors using the Term Frequency-Inverse Document Frequency (tf-idf) from *SciKit-Learn* with parameter *sublinear\_tf=True*, *min\_df=5*, *norm='l2'*, *encoding='latin-1'*, *ngram\_range=(1, 2)*, *stop\_words='english'*. The obtained number of features was 3166. While the parameters of SSVM were *sublinear\_tf=True*, *analyzer='word'*, *tokenizer=tokenize*, *lowercase=True*, *ngram\_range=(1, 1)*, *stop\_words=en\_stopwords*, *norm='l2'*, *min\_df=3*. The obtained number of features was 3844.

The series of VLGCA-NB preprocessing was different from other classification methods because it did not need the vector form of words. All words were converted to lowercase. Number, stop words, punctuation, non-English words, non-alphabetic characters, and words smaller than two characters were removed, lemmatization was performed, and all contractions were replaced by their real words or phrases. The number of features was decreased to 493 after the preprocessing step.

### 5.2 Parameters

To apply CNN to the classification, we used fixed parameters with *maximumEpoch = 10* and *batchSize = 100*. We also employed fixed parameters for VLGCA-NB with *PopulationSize = 100*, *Generationsize = 50*, *Crossoverrate = 0.7*, and *Mutationrate = 0.3*.

RF, CNN, and VLGCA-NB are stochastic algo-

rithms which mean the results will differ for each run. Therefore, those algorithms were run five times. The best result among the five attempts was selected for comparison with other algorithms.

### 5.3 Evaluation Measurement

Classifier performance evaluation using accuracy is often considered as a suited measurement. However, the datasets from both subtasks were imbalanced. Majority class is often reckoned by the classifier, thus, higher accuracy will be achieved for it. Therefore, in this research, we used Precision, Recall, and F1-Score as the main evaluation measures. Accuracy measurement (Equation (1)) was still used in the fitness function of VLGCA-NB. F1-Score (Equation (4)) computation relied on Precision (Equation (2)) and Recall (Equation (3)) measurements. As the Suggestion results were more concerned, the evaluation of this competition was the Suggestion results' F1-Score.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1-Score = 2x \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

where :

*TP* = True Positive

*TN* = True Negative

*FP* = False Positive

*FN* = False Negative

Evaluation measurement for VLGCA-NB was done in every generation using Fitness Function based on the result of Naive Bayes classification.

The Fitness value was found by addition of accuracy, F1-Score of Suggestion and F1-Score of Non-suggestion, which were defined as follows:

$$Fitness = Accuracy + F1-Score_{suggestion} + F1-Score_{non-suggestion} \quad (5)$$

## 6 Results and Discussion

In this section, we evaluated the classification performance of the seven classification methods with the baseline for both Subtasks. The methods performance was evaluated on Precision, Recall, and F1-Score metrics, except for the VLCGA-NB, we still used accuracy in the fitness function.

A typical observation from the confusion matrix produced in this research was that the number of correct classification was higher than the number of misclassified for the Non-suggestion class, except for baseline of Subtask A. This result was unvaried across all methods, with a little difference of the whole classification count. Though Subtask A aimed to classify Suggestion in the same domain, the number of correct classification of Suggestion class was lower than the number of misclassified for most of the classification methods. In particular, baseline and SSVC obtained better results than other utilized methods. Furthermore, as Subtask B aimed to classify Suggestion in the different domain, eventually it was hard for all classification methods to obtain even fair results. All of them failed to obtain a higher number of correct classification of the Suggestion class.

Table 2 shows the precision, recall, and F1-Score comparisons for each class in Subtask A. We noted that the obtained result of all classification methods outperformed that of the baseline for the Non-suggestion class. MNB yielded the best results with 0.95.

In terms of F1-Score of the Suggestion class, refer to Table 2, we noted that RF, SSVC, and VLCGA-NB obtained a competitive result outperforming baseline for Suggestion class. The highest F1-Score was obtained by SSVC at 0.47. RF and VLCGA-NB produced F1-Score at 0.29 and 0.31, respectively. Overall, note that MNB and SSVC obtained the best F1-Score for Non-suggestion and Suggestion classes, respectively.

Table 3 shows the experimental results for the dataset in Subtask B. We noted that the F1-Score of Non-suggestion yielded a good result with the higher F1-Score obtained by RF classifier. However, the F1-Score of Suggestion class produced

poor finding, except for the baseline. F1-Score of Suggestion class using the baseline achieved surprisingly well considering their simplicity, which yielded 0.73. A possible reason for this finding might be that the manually selected keywords and patterns based on which usually used to suggest something would make use of the common Suggestion sentence that a machine might not be able to discover. A possible problem with the baseline approach was probably that the keywords and patterns for Suggestion class might be also used for Non-suggestion class. Therefore, it might be difficult for baseline to define which keywords and patterns actually used in Suggestion sentences. This could be proven from the F1-Score results for the Non-suggestion class in Subtask A which yielded the lowest result of 0.59.

Regarding the number of features selected by VLCGA-NB, the features were decreased from 493 to 372. Refer to our defined expectation in 4.3, VLCGA-NB was able to produce a limited number of features which has major contribution to the Suggestion classification. This could be proven from its F1-Score results which yielded higher value compared to the baseline in Subtask A.

## 7 Conclusion

This paper has described our approach for participating in both Subtask A and Subtask B of Task 9 of SemEval-2019, that is, suggestion mining from online reviews and forum (Negi et al., 2019). Our approach explored and compared various classification methods, namely, Random Forest, Logistic Regression, Multinomial NB, Linear SVC, Sub-linear SVC, CNN, and VLCGA-NB. Since the datasets provided by the organizer were imbalanced data, it was more important to correctly classify a sentence as a Suggestion class. Thus, the F1-Score of the Suggestion class was more considered. Compared to the baseline, all algorithms performed better classification for the Non-suggestion class in both Subtask A and Subtask B. In contrast, they performed worse classification than the baseline for the Suggestion class in both Subtask A and Subtask B. This poor performance was except for the RF, SSVC, and VLCGA-NB that could outperform the baseline for classifying the Suggestion class in Subtask A. Based on our results, we observed that besides the imbalanced data, the implicit meaning problem related to the

Table 2: Metrics Report Subtask A

Method	Precision		Recall		F1-Score	
	Non suggestion	Suggestion	Non suggestion	Suggestion	Non suggestion	Suggestion
<b>Baseline</b>	0.98	0.16	0.42	0.92	0.59	0.27
<b>Random Forest</b>	0.92	0.34	0.94	0.25	0.93	0.29
<b>Logistic Regression</b>	0.9	0.36	0.98	0.1	0.94	0.16
<b>Linear SVC</b>	0.91	0.21	0.87	0.3	0.89	0.25
<b>Sublinear SVC</b>	0.97	0.35	0.84	0.75	0.9	0.47
<b>Multinomial NB</b>	0.9	0.62	1	0.06	0.95	0.11
<b>CNN</b>	0.9	0.08	0.98	0.01	0.94	0.02
<b>VLCGA-NB</b>	0.92	0.45	0.97	0.23	0.94	0.31

Table 3: Metrics Report Subtask B

Method	Precision		Recall		F1-Score	
	Non suggestion	Suggestion	Non suggestion	Suggestion	Non suggestion	Suggestion
<b>Baseline</b>	0.82	0.69	0.74	0.78	0.78	0.73
<b>Random Forest</b>	0.59	0.49	0.88	0.15	0.93	0.29
<b>Logistic Regression</b>	0.58	0.41	0.95	0.05	0.72	0.08
<b>Linear SVC</b>	0.6	0.55	0.9	0.17	0.72	0.26
<b>Sublinear SVC</b>	0.64	0.68	0.9	0.29	0.74	0.37
<b>Multinomial NB</b>	0.57	0.13	0.97	0.01	0.72	0.01
<b>CNN</b>	0.58	0.51	0.96	0.05	0.73	0.1
<b>VLCGA-NB</b>	0.6	0.69	0.96	0.11	0.74	0.19

Suggestion class was also the challenge of the suggestion mining. The feature selection corresponds with the Suggestion class will be our future intention. In addition, it might be valuable to inspect further the use of our approach to other text classification tasks such as deceptive opinions (Siagian and Aritsugi, 2017, 2018) and fake news identifications.

## Acknowledgments

The authors acknowledge the anonymous reviewers for their valuable comments and suggestions regarding this paper. A. H. A. M. Siagian thanks the scholarship support from Riset-Pro (Research and Innovation in Science and Technology Project) KEMENRISTEKDIKTI (Ministry of Research, Technology and Higher Education of the Republic of Indonesia).

## References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever,

Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. *TensorFlow: Large-scale machine learning on heterogeneous systems*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O’Reilly Media.

Caroline Brun and Caroline Hagege. 2013. *Suggestion mining: Detecting suggestions for improvement in users comments*. volume 70, pages 199–209. Research in Computing Science.

François Chollet et al. 2015. Keras. <https://keras.io>.

François Chollet. 2017. Using pre-trained word embeddings in a Keras model. In *The Keras Blog*. <https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.html>.

Li Dong, Furu Wei, Yajuan Duan, Xiaohua Liu, Ming Zhou, and Ke Xu. 2013. *The automated acquisition of suggestions from tweets*. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI’13, pages 239–245. AAAI Press.

Tirana Noor Fatyanosa. 2019a. Neural network for text classification. <https://github.com/TiraNosa/NeuralNetworkForTextClassification>.

- Tirana Noor Fatyanosa. 2019b. Text classification using vlcca-nb. <https://github.com/TiraNosa/Text-Classification-using-VLCCA-NB>.
- Tirana Noor Fatyanosa. 2019c. Text classification with scikit-learn. <https://github.com/TiraNosa/Text-Classification-with-Scikit-Learn>.
- Tirana Noor Fatyanosa, Fitra A. Bachtiar, and Mahendra Data. 2018. Feature Selection using Variable Length Chromosome Genetic Algorithm for Sentiment Analysis. In *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*, Malang.
- Andrew B. Goldberg, Nathanael Fillmore, David Andrzejewski, Zhiting Xu, Bryan Gibson, and Xiaojin Zhu. 2009. May All Your Wishes Come True: A Study of Wishes and How to Recognize Them. In *The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 263–271.
- J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95.
- Yang-Yin Lee, Hao Ke, Hen-Hsen Huang, and Hsin-Hsi Chen. 2016. Combining word embedding and lexical database for semantic relatedness measurement. In *Proceedings of the 25th International Conference Companion on World Wide Web, WWW '16 Companion*, pages 73–74, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.
- Wes McKinney. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, pages 51 – 56.
- Sapna Negi. 2016. Suggestion mining from opinionated text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics – Student Research Workshop*, pages 119–125.
- Sapna Negi, Kartik Asooja, Shubham Mehrotra, and Paul Buitelaar. 2016. A study of suggestions in opinionated texts and their automatic detection. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 170–178.
- Sapna Negi and Paul Buitelaar. 2015. Towards the extraction of customer-to-customer suggestions from reviews. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2159–2167.
- Sapna Negi and Paul Buitelaar. 2017. Inducing distant supervision in suggestion mining through part-of-speech embedding. arXiv:1709.07403. Version 2.
- Sapna Negi, Tobias Daudert, and Paul Buitelaar. 2019. SemEval-2019 Task 9: Suggestion mining from online reviews and forums. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*.
- Sapna Negi, Maarten de Rijke, and Paul Buitelaar. 2018. Open domain suggestion mining: Problem definition and datasets. arXiv:1806.02179. Version 2.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Janardhanan Ramanand, Krishna Bhavsar, and Niranjan Pedanekar. 2010. Wishful thinking: Finding suggestions and 'buy' wishes from product reviews. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, CAAGET '10*, pages 54–61, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Al Hafiz Akbar Maulana Siagian and Masayoshi Aritsugi. 2017. Combining word and character n-grams for detecting deceptive opinions. In *2017 IEEE 41st Annual Computer Software and Applications Conference, COMPSAC*, pages 828–833, Washington, DC, USA. IEEE Computer Society.
- Al Hafiz Akbar Maulana Siagian and Masayoshi Aritsugi. 2018. Exploiting function words feature in classifying deceptive and truthful reviews. In *2018 Thirteenth International Conference on Digital Information Management, ICDIM*, pages 51–56, Washington, DC, USA. IEEE Computer Society.
- Alfan Farizki Wicaksono and Sung-Hyon Myaeng. 2012. Mining advices from weblogs. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12*, pages 2347–2350, New York, NY, USA. ACM.
- Alfan Farizki Wicaksono and Sung-Hyon Myaeng. 2013. Automatic extraction of advice-revealing sentences for advice mining from online forums. In *Proceedings of the Seventh International Conference on Knowledge Capture, K-CAP '13*, pages 97–104, New York, NY, USA. ACM.